


# Using

## Forms to Control Input

---

### Objectives

- ▶ Plan a form
- ▶ Create a text entry field
- ▶ Add radio buttons
- ▶ Add checkboxes
- ▶ Create a pull-down menu
- ▶ Add a push button
- ▶ Connect a form's back end

While the Web is an effective means to make information available, HTML also includes interactive aspects. On many Web sites, in addition to reading information, users can submit information as well. The main HTML element that facilitates interactivity is called a form. A **form** is the set of Web page elements with which you interact to submit a query to a search engine or provide credit card and shipping information when you make purchases on the Web.  Grace Dekmejian works in the Information Systems department at Nomad Ltd, a travel and sporting goods company. Nomad is forming a corporate division to handle online sales. Along with her co-workers, Grace is creating a company Web site. The next task for Grace and her Web development team is to create a Web page form that allows potential customers to provide the necessary information to complete their orders.



# Planning a Form

Including a form on your Web site is a simple and straightforward way to allow your site's users to send you feedback or other information. If you are using the Web to sell products or services, forms are vital. As shown in Figure D-1, a Web form can allow users to enter information, but it also can provide predefined choices from which users can select. Most forms include several **fields**, which are the form elements, such as text boxes or pull-down menus, that allow user input. Each field or group of fields is usually associated with a **label**, which is the text that explains what information is required by the adjacent field. Grace has met with her co-workers who are designing the Nomad online store, to determine the information users need to provide when making an online purchase. Grace reviews the different types of form fields available in HTML, and selects the most appropriate field for each type of information. Figure D-2 shows the sketch of Grace's planned form, she plans to create, which includes the following types of fields:

## Details



### Single-line text box

A single-line text box is ideal for requesting limited user input, such as first or last name, or street address. HTML allows you to specify the size of a single-line text box, as well as the maximum character length of user input allowed.



### Checkboxes and radio buttons

Checkboxes and radio buttons simplify Web forms by allowing users to select from a list of options. Users can select multiple checkboxes, which are ideal for a set of options from which a user might choose either none, one, or several—for example, pizza toppings. On the other hand, within a set of radio buttons, a user can select only one. Radio buttons are the best choice for a mutually exclusive set of choices, such as t-shirt size (S, M, L, or XL).



### Pull-down menus and scroll boxes

Pull-down menus and scroll boxes both serve a similar function to radio buttons: they allow users to make one selection from among multiple items. When your list of choices is lengthy, pull-down menus and scroll boxes conserve Web page space. A pull-down menu shows one choice by default, and displays the remaining choices after a user clicks the down arrow. By contrast, a scroll box always shows the same number of choices, which is customizable. Users view the hidden options by clicking an arrow and scrolling through the list.



### Multi-line text areas

Multi-line text areas are designed to allow less-structured input by the user. Such an area is often used to allow additional comments or questions at the end of a form. You can specify the dimensions of this area on the page.



### Push buttons

HTML includes code to create two kinds of predefined buttons that users can click to execute common form-related tasks. You can configure one of these buttons to submit the information entered by users to the appropriate processing system. The other button allows users to clear information they have entered and start over.

**FIGURE D-1:** Web page allowing user input with a form

Predefined choices  
that users can select

☒ Standard Shipping  
☐ 2 Day Air Shipping  
☐ Overnight Air Shipping

**3. Packaging Options**

☐ Gift wrap (\$2.00 per item)  
☐ Remove price tags (free)  
☐ Ship multiple items together

**4. Payment Information**

Credit Card:

Card Number:

Expiration Date (mm/yyyy):

☐ Billing address same as ship-to address

Text boxes in  
which users can  
enter information

**FIGURE D-2:** Grace's sketch of the Nomad Ltd checkout form

Nomad logo      Checkout Form

1. Ship-to Address

text forms for  
address information

2. Shipping Method

radio buttons to allow  
selection of only one option

3. Packaging Options

checkboxes to allow  
multiple selections

4. Payment Information

-pull-down menu for credit card options  
-text boxes for expiration date and card number  
-checkbox for same billing address

submit button  
displaying text  
"Place Order"



## Web users and privacy

Surveys consistently show that most Web users are concerned that their personal information is not adequately secure when they interact on the Web. The topic of security is important when considering whether your Web site will include forms which are often designed to collect information that people want to keep as private as possible. A Web designer is not often in a position to decide a company's policy on use of personal information collected from Web users. You can, however, make users of your site feel more secure by clearly communicating your

organization's policy regarding use of the information you request. Nearly all Web sites that request information allow users to select options within the form that keep their information off e-mail and printed material mailing lists. Finally, sometimes organizations want to collect additional information if users are willing to provide it. Be sure your Web page makes it clear that such fields are optional; then you avoid losing the business of potential customers who refuse to answer such questions.



# Creating a Text Entry Field

In HTML, the text and fields that compose a form are always surrounded by the `<FORM>..</FORM>` tag pair. Within a form, you add most common fields with the `<INPUT>` tag, and use the `TYPE` attribute to specify the kind of field you want to create. To create a text entry field, you set the `TYPE` attribute value to “text”. Table D-1 explains the syntax and commonly-used attributes for the `<FORM>..</FORM>` and `<INPUT>` tags. Text entry fields are useful for collecting basic information in most forms, which usually include the user’s name and address. Grace has already created the HTML document structure for the form, and has inserted headings for the various sections of the form. She also added comments indicating where each form element should be placed. She begins coding the form by adding the `<FORM>..</FORM>` tag pair to her document. She uses, the `<INPUT>` tag to create the text fields and labels for shipping and billing information.

## Steps 1 2 3 4

### QuickTip

The `<!-->` tag indicates a Web page designer’s comment; text enclosed in the tag does not appear in the browser.

1. Start your text editor program, then open the file **HTM D-1.htm**, and save it as a text document with the filename **nomad-d.htm**

2. Select the text **`<!-- BEGIN FORM -->`** near the top of the document, press **[Delete]**, then type **`<FORM NAME="checkout">`**

Although Grace doesn’t expect to immediately make use of the `NAME` values, she plans to include the `NAME` attribute with each of her form elements to facilitate future enhancements to the form.

3. Move the insertion point near the bottom of the document, select the text **`<!-- END FORM -->`**, press **[Delete]**, then type **`</FORM>`**

Grace will enter code for all her form fields between these beginning and ending tags.

4. Move the insertion point near the top of the document, select the text **`<!-- SHIPPING ADDRESS - TEXT ENTRY FIELDS -->`**, then press **[Delete]**

### QuickTip

When writing HTML, it’s good practice to indent multiple lines that are nested within a tag pair.

5. Type **`First Name: <INPUT TYPE="text" SIZE="20" NAME="shipfirstname">`**, press **[Enter]**, press **[Spacebar]** four times, type **`<BR>`**, then press **[Enter]**

This step creates the label and associated field where the user will enter the first name of the order’s recipient. The `SIZE` attribute specifies the width of the text box. Just like the `NAME` attribute associated with the `<FORM>` tag in Step 2, including `NAME` for each form element makes it easier to add advanced features to your form in the future.

6. Enter the remaining code for the shipping information section, as shown in Figure D-3  
This portion of your completed HTML document should look like the one shown in Figure D-3, although your lines may wrap differently.

7. Save your work, start your Web browser, then open the file **nomad-d.htm** in your browser

The text entry boxes and labels appear beneath the heading “1. Ship-to Address,” as shown in Figure D-4.

8. Click in the **First Name text box**, type your first name, press **[Tab]**, type your last name in the **Last Name text box**, then continue to press **[Tab]** to complete the remaining fields using your own (or fictitious) information

Grace tests her fields to be sure that they allow input as expected, and that the length of each field is appropriate.

FIGURE D-3: Web page code containing form and text box tags

Opening  
<FORM> tag

Form field labels

```
<FORM NAME="checkout">
  <IMG SRC="images/sta.gif" WIDTH="180" HEIGHT="28" ALT="Ship-to Address">
  <BR>
  <EM>(optional fields in italic)</EM>
  <P>
    First Name: <INPUT TYPE="text" SIZE="20" NAME="shipfirstname">
    <BR>
    Last Name: <INPUT TYPE="text" SIZE="20" NAME="shiplastname">
    <BR>
    Address Info.: <INPUT TYPE="text" SIZE="40" NAME="shipaddr1">
    <BR>
    <EM>Address Line 2</EM>: <INPUT TYPE="text" SIZE="40" NAME="shipaddr2">
    <BR>
    City: <INPUT TYPE="TEXT" SIZE="20" NAME="shipcity">
    <BR>
    State: <INPUT TYPE="TEXT" SIZE="2" MAXLENGTH="2" NAME="shipstate">
    <BR>
    Zip Code: <INPUT TYPE="TEXT" SIZE="10" MAXLENGTH="10" NAME="shipzip">
    <BR>
    <EM>Phone</EM>: ( <INPUT TYPE="TEXT" SIZE="3" MAXLENGTH="3"
NAME="shipareacode"> )
    <INPUT TYPE="TEXT" SIZE="3" MAXLENGTH="3"
NAME="shipphoneprefix"> -
    <INPUT TYPE="TEXT" SIZE="4" MAXLENGTH="4"
NAME="shipphoneend">
    <BR>
    E-mail: <INPUT TYPE="TEXT" SIZE="25" NAME="email">
```

<INPUT> tags for  
text entry boxes

FIGURE D-4: Web page displaying labeled text boxes


Labels

Text entry boxes

TABLE D-1: Common form field tags

tag(s)	attribute	description	example
<FORM>..</FORM>	NAME	creates a form name; used to easily reference a form when adding advanced features	<FORM NAME= "checkout">..</FORM>
<INPUT>	TYPE	specifies the type of form field to create	<INPUT TYPE="text">
	SIZE	specifies the display length of the form field	<INPUT SIZE="20">
	MAXLENGTH	sets the largest number of characters allowed in the current form field; extra typing is ignored after this number is reached	<INPUT MAXLENGTH="40">

# Adding Radio Buttons

Text boxes are ideal tools for allowing form users to enter the information unique to each person. But, in many other situations, you can reduce the potential for user input error by providing a list of choices from which the user can select. This eliminates the possibility that a user will enter a selection that the order-processing system doesn't understand. HTML includes several list formats for forms; each format is appropriate for different situations and types of information. To create a set of options that a user can see all at once, and from which a user can make only one selection, you create radio buttons. **Radio buttons** are small white circles, each appearing next to explanatory text. When a user clicks a circle, it fills with black to designate it as selected. You create radio buttons using the `<INPUT>` tag, with the `TYPE` attribute set to "radio." Table D-2 explains the `<INPUT>` attributes commonly used when creating radio buttons.  Nomad offers its customers a choice between three shipping speeds. Grace adds these to the form as a set of radio buttons.

## Steps 1 2 3 4

1. In your text editor, select the text `<!-- SHIPPING METHOD - RADIO BUTTONS -->` near the middle of the document
2. Press **[Delete]**, then type `<INPUT TYPE="radio" NAME="shipvia" VALUE="standard" CHECKED>Standard Shipping`, then press **[Enter]**  
Unlike most form elements, to work properly, a set of radio buttons requires the `NAME` attribute. HTML uses this attribute to identify the radio buttons that you group together. The `NAME` attribute defines the set from which users can make only one selection. The `CHECKED` attribute signifies that the radio button should appear as selected when the page opens in the browser.
3. Press **[Spacebar]** four times, type `<BR>`, then press **[Enter]**
4. Press **[Spacebar]** four times, type `<INPUT TYPE="radio" NAME="shipvia" VALUE="priority">2 Day Air Shipping`, then press **[Enter]**  
Each remaining item in the radio button list uses the same `TYPE` and `NAME` settings as the first item, but has a different `VALUE` and label text.
5. Press **[Spacebar]** four times, type `<BR>`, then press **[Enter]**
6. Press **[Spacebar]** four times, then type `<INPUT TYPE="radio" NAME="shipvia" VALUE="overnight"> Overnight Air Shipping`  
Your HTML document should look like the one shown in Figure D-5.
7. Save your work, click the **browser program button** on the taskbar, reload the form in the browser window, then scroll down if necessary to view the radio buttons  
The three labeled radio buttons appear as a group, as shown in Figure D-6.
8. If necessary, click the **Standard Shipping radio button**, click the **2 Day Air Shipping radio button**, then click the **Overnight Air Shipping radio button**  
When you click a radio button, a black dot appears in the center, indicating that it is selected. When a radio button is selected and you click another radio button, only the most recently clicked button is marked as selected.



**FIGURE D-5:** Web page code containing tags for radio button list

<INPUT>  
tags for  
radio  
buttons

```

<BR><BR>

<IMG SRC="images/sm.gif" WIDTH="200" HEIGHT="28" ALT="Shipping Method">
<P>
  <INPUT TYPE="radio" NAME="shipvia" VALUE="standard" CHECKED>Standard
Shipping
  <BR>
  <INPUT TYPE="radio" NAME="shipvia" VALUE="priority">2 Day Air Shipping
  <BR>
  <INPUT TYPE="radio" NAME="shipvia" VALUE="overnight">Overnight Air
Shipping
  <BR><BR>

```

Radio button labels

**FIGURE D-6:** Web page displaying radio button list

Radio  
button list

E-mail:

## 2. Shipping Method

- ☒ Standard Shipping
- ☐ 2 Day Air Shipping
- ☐ Overnight Air Shipping

## 3. Packaging Options

**TABLE D-2:** Radio button <INPUT> tag attributes

attribute	function	example
<b>TYPE</b>	setting a value to "radio" identifies the form element as a radio button	<INPUT TYPE="radio">
<b>NAME</b>	assigns a name to the associated radio buttons, identifying them as a group	<INPUT TYPE="radio" NAME="shipvia">
<b>VALUE</b>	Specifies the text to be submitted to the order- processing system if the current radio button is selected	<INPUT TYPE="radio" VALUE="priority">
<b>CHECKED</b>	marks the current radio button to appear in the browser as selected when page opens; only one radio button per set may include this attribute	<INPUT TYPE="radio" CHECKED>




## More about the CHECKED attribute

By default, all the fields in a set of radio buttons appear empty. This means that none of the buttons is selected unless the user clicks one of them. You can include the CHECKED attribute in the <INPUT> tag for one of the radio buttons in a set, causing it to appear as selected when the Web page opens. The CHECKED attribute can simplify the form for your

users by saving them a click. It's most useful for the least expensive or most commonly selected items in a list. In general, you use the CHECKED attribute in a list that requires a selection by the user; you avoid using it in a list which doesn't require that a user make a selection.

# Adding Checkboxes

While radio boxes allow users to make a single choice from a set of options, sometimes you want to allow users to make multiple selections from a list. For situations like this, HTML offers **checkboxes**, which display an array of choices that are all visible at once, and from which users may select any, all, or none. A checkbox appears as a small white box next to its label. When a user clicks an empty checkbox, a check mark appears in the box; clicking a checked box removes the check mark. You create checkboxes using the `<INPUT>` tag, with the `TYPE` attribute set to “checkbox.” Unlike radio buttons, each checkbox should have a unique `NAME` value. Additionally, code for checkboxes does not require a `VALUE` attribute, and multiple checkboxes may use the `CHECKED` attribute to display a check mark by default.  Nomad Ltd offers customers several choices regarding how their purchases are packaged. Because these choices are not exclusive of each other, Grace adds them to the form as checkbox fields.

## Steps 1234

1. In your text editor, select the text `<!--PACKAGING OPTIONS - CHECKBOXES -->`, then press **[Delete]**
2. Type `<INPUT TYPE="checkbox" NAME="wrap">Gift wrap ($2.00 per item)`, then press **[Enter]**  
This code creates a checkbox field with the name “wrap,” and adds a descriptive label next to it. Because all the items in her checkbox list are optional, Grace decides not to use the `CHECKED` attribute.
3. Press **[Spacebar]** four times, type `<BR>`, then press **[Enter]**
4. Press **[Spacebar]** four times, type `<INPUT TYPE="checkbox" NAME="tags">Remove price tags (free)`, then press **[Enter]**
5. Press **[Spacebar]** four times, type `<BR>`, then press **[Enter]**
6. Press **[Spacebar]** four times, then type `<INPUT TYPE="checkbox" NAME="together">Ship multiple items together`  
Your HTML document should look like the one shown in Figure D-7.
7. Save your work, click the **browser program button** on the taskbar, reload the form in the browser window, then scroll down if necessary to view the checkbox list  
As Figure D-8 shows, the three labeled checkboxes are grouped under the heading “Packaging Options.”
8. Click each checkbox to check it, then click each checkbox again to remove the check marks  
Clicking an empty checkbox adds a check mark, while clicking a checked checkbox removes the check mark. This is known as a **toggle**.



**FIGURE D-7:** Web page code containing tags for checkbox list

```
<BR><BR>

<IMG SRC="images/po.gif" WIDTH="221" HEIGHT="28" ALT="Packaging Options">
<P>
  <INPUT TYPE="checkbox" NAME="wrap">Gift wrap ($2.00 per item)
  <BR>
  <INPUT TYPE="checkbox" NAME="tags">Remove price tags (free)
  <BR>
  <INPUT TYPE="checkbox" NAME="together">Ship multiple items together
<BR><BR>

<IMG SRC="images/pi.gif" WIDTH="249" HEIGHT="28" ALT="Payment Information">
```

<INPUT> tags for  
checkboxes

Checkbox labels

**FIGURE D-8:** Web page displaying checkbox list

☐ 2 Day Air Shipping  
☐ Overnight Air Shipping

### 3. Packaging Options

☐ Gift wrap (\$2.00 per item)  
☐ Remove price tags (free)  
☐ Ship multiple items together

### 4. Payment Information

Checkbox list




### Labeling radio buttons and checkboxes

The text that you enter as a label next to a radio button or a checkbox is just like plain text elsewhere on your Web page. Even though form fields would be meaningless to users without labels, HTML has no rules about how to label the fields. Therefore, you're free to put a label to the left or right of a field, or even above or

below it. The most important guideline is to make sure it's obvious to a user which label and field go together. From a design standpoint, it's also important to make the fields easy for a user to find. This is why most Web designers align the fields on Web page forms, rather than aligning the field labels.

# Creating a Pull-Down Menu

Radio buttons and checkboxes lay out a group of related options so that they all are visible on the Web page. Sometimes, however, you may want to save space on your Web page, or you may have such an extensive set of choices that it would be impractical to include all of them in the Web page's layout. For this type of situation, HTML allows you to format your options with a more efficient form field called a pull-down menu. A **pull-down menu**—or **drop-down list**—appears on the page like a single-line text field, but also contains a button marked with an arrow. When a user clicks the button, a menu of choices opens. When the user clicks a choice, the list closes, and the user's selection appears in the field. The pull-down menu is popular on Web page forms because it uses space efficiently, and because most users are already familiar with it from common operating systems and software. To create a pull-down menu, you use the `<SELECT>..</SELECT>` tag pair to surround the list of choices. Each choice is marked with the `<OPTION>` tag. Table D-3 explains the common attributes used with these tags.

 The last section in Grace's planned form allows customers to enter credit card information. Because Nomad Ltd accepts six different types of credit cards, Grace wants to list them in a pull-down menu to save space.

## Steps 1234

1. In your text editor, select the text `<!-- CREDIT CARD - PULL-DOWN MENU -->`, press **[Delete]**, type **Credit Card:**, then press **[Enter]**
2. Press **[Spacebar]** six times, type `<SELECT NAME="card">`, then press **[Enter]**  
The `<SELECT>` tag marks the start of the pull-down menu.
3. Press **[Spacebar]** eight times, type `<OPTION VALUE="v">Visa`, then press **[Enter]**
4. Repeat Step 3 to enter the remaining five options for the credit card pull-down menu, shown in Figure D-9
5. Press **[Spacebar]** six times, type `</SELECT>`, then press **[Enter]**  
The `</SELECT>` tag marks the end of the pull-down menu list.

### QuickTip

Many Web pages also use pull-down menus to allow users to select the month and year of a credit card expiration date, rather than typing it.

6. Replace the comment `<!-- CARD INFO - TEXT ENTRY FIELDS + CHECKBOX -->` with the code shown in Figure D-9  
Your Web document should match the one shown in Figure D-9.
7. Save your work, click the **browser program button** on the taskbar, reload the form in the browser window, then scroll down to view the pull-down menu  
As shown in Figure D-10, the Web page displays a pull-down menu field in the "Payment Information" section.
8. Click the **Credit Card pull-down arrow**, then click one of the options on the menu that opens  
The option you clicked now appears in the pull-down menu box.

**FIGURE D-9:** Web page code containing tags for pull-down menu

```

<BR><BR>

<IMG SRC="images/pi.gif" WIDTH="249" HEIGHT="28" ALT="Payment Information">
<P>
  Credit Card:
  <SELECT NAME="card">
    <OPTION VALUE="u">Visa
    <OPTION VALUE="m">MasterCard
    <OPTION VALUE="a">American Express
    <OPTION VALUE="i">Diners Club
    <OPTION VALUE="d">Discover
    <OPTION VALUE="j">JCB
  </SELECT>
  <BR>
  Card Number:
  <INPUT TYPE="text" SIZE="20" MAXLENGTH="20" NAME="cardno">
  <BR>
  Expiration Date (mm/yyyy):
  <INPUT TYPE="text" SIZE="7" MAXLENGTH="7" NAME="cardexp">
  <BR><BR>
  <INPUT TYPE="checkbox" NAME="sameaddr">Billing address same as ship-to
  address
  <BR><BR>

```

Pull-down menu code

Code to replace CARD INFO comment

**FIGURE D-10:** Web page displaying pull-down menu

4. Payment Information

Credit Card: Visa

Card Number:

Expiration Date (mm/yyyy):

☐ Billing address same as ship-to address

Pull-down menu displaying default menu choice

Click to display menu

Additional Payment Information fields


**TABLE D-3:** Pull-down menu tags and attributes

tag(s)	function	attribute	description
<SELECT>.. </SELECT>	creates a list field, in which some choices are hidden; start and end tags surround list options, which are marked with <OPTION>	NAME	creates a field name; used to easily reference a field when adding advanced features
		SIZE	sets the number of items which appear by default; if omitted, choices appear as a pull-down menu; if set to 2 or more, list appears as a scroll box
		MULTIPLE	sets the number of options user may select from the list, and automatically formats list as a scroll box; if omitted, user may select only one list item
<OPTION>	precedes each item in a list field; must occur between <SELECT> and </SELECT> tags	VALUE	specifies the text to be submitted to order-processing system if the current option is selected; optional
		SELECTED	specifies the current option to appear in the field text box by default when the Web page opens in the browser





# Adding a Push Button

Occasionally, while completing Web page forms, users need the browser to perform an action on the form. Most users commonly need to indicate that they are finished with the form, and they need to submit their information to your organization for processing. To allow users to execute tasks, HTML forms include **push buttons**, which are labeled objects that a user can click to perform a task. You create a push button using the `<INPUT>` tag, and you specify its functions with different attributes. HTML includes two predefined functions, reset and submit, that you can assign to buttons by using the `TYPE` attribute. A **reset button** clears all the input in a form, allowing a user to start over. Reset buttons are no longer common on the Web, since forms today are generally easy to edit. The **submit button** performs one of the most common form activities: submitting information for processing. Table D-4 explains basic `<INPUT>` attributes for creating a submit button.  According to her sketch, Grace needs to add just one more element to her form: a submit button.

## Steps 1234

1. In your text editor, select the text `<!-- SUBMIT BUTTON -->` near the bottom of the Web document, then press **[Delete]**
2. Type `<INPUT TYPE="submit" VALUE="Place Order" NAME="order">`  
This portion of your HTML document should look like the document shown in Figure D-11.
3. Save your work, then click the **browser program button** on the taskbar
4. Reload the form in the browser window, then scroll down to view the submit button  
Grace added a submit button code to create a shaded rectangular button with the text “Place Order,” as shown in Figure D-12.
5. Click the **Place Order button**  
The browser reloads the page and clears the information you entered in the form. Once the Nomad Ltd online ordering system is functional, it will respond to the submission by opening a different Web page, confirming that the user’s order was received.

**TABLE D-4:** Basic `<INPUT>` attributes for push buttons

attribute	description
TYPE	<b>submit</b> creates a button for sending entered data to the organization for processing <b>reset</b> clears all the fields in the current form <b>button</b> creates a generic button that you can customize using advanced Web authoring skills <b>image</b> uses a graphic (specified by the SRC attribute) as the button
NAME	creates a field name; used to easily reference a field when adding advanced features
VALUE	defines the text that appears on the button, as well as the value submitted to the order-processing system when the current button is pushed

**FIGURE D-11:** Web page code containing tag for push button

```
<BR><BR>
<INPUT TYPE="submit" VALUE="Place Order" NAME="order">
</FORM>
</BODY>
```

<INPUT>  
tag for  
submit  
push button

**FIGURE D-12:** Web page displaying push button

☐ Billing address same as ship-to address

Place Order

Customized  
text set  
by VALUE  
attribute



### Using a graphic as a push button

The VALUE attribute allows you to customize the text on a push button to fit its specific function on your Web pages. You can further customize a button to fit your page's overall theme by specifying a graphic to use as a button. To use a graphic as a push button, set

the <INPUT> tag TYPE attribute to "image," and specify the path and name of the graphic file with the SRC attribute. Like other images in your Web pages, you should specify alternate text for a graphical button by using the ALT attribute.

**FIGURE D-13:** Web page using a graphic as push button

Text box used to enter search terms

Graphic serves as submit button to begin search

# Connecting a Form's Back End

Like tags to format other Web page elements such as text and graphics, the HTML tags for creating a form simply generate the Web page that users see in their browsers; that page is known as the **front end**. Once a user clicks the submit button, however, the information entered requires processing that is not performed by the page's HTML code. When a submit button is correctly programmed, it instructs the browser to send the entered data to your Web server for processing. The programs that reside on an organization's computer system and are responsible for processing the submitted data make up the **back end**. Figure D-14 shows the different parts of the front end and the back end, and how data moves between them. Generally, Web page designers are responsible for creating the front end, while other people in an organization, such as the Web site administrator, handle the back end tasks. However, it's important that the front-end designer work with the people responsible for the back end to ensure that the parts of the Web site that both of you create will work together as expected. Grace is meeting with Cleveland Mack, the Nomad Ltd Web site administrator, to begin coordinating their work on the checkout process for the online store. Cleveland explains the process of submitting information entered in a form:

## Details



### Browser packaging

When a user clicks the submit button, the browser refers to the <FORM> tag attributes for details on how to submit the information. These attributes, which are described in Table D-5, allow you to specify where the information entered in a form is sent, how the browser sends it, and in what format. Cleveland has not yet finalized the submission requirements, but will provide Grace with these settings in time to test the system before the checkout page goes into use.



### CGI

The border between a Web site's front end and back end is the communication between the Web server—where the HTML document is stored—and the programs that process the information collected in a form. This communication takes place using a standardized protocol known as **CGI (Common Gateway Interface)**. The target on the Web server for the form contents is a file containing a short set of instructions, known as a script, which uses CGI to specify how the data is sent to the back end. Once you know how to create a Web page in HTML, learning to create the script for a form is generally an elementary task. However, since Grace is creating her first Web site, Cleveland offers to take responsibility for the script.



### Order processing and confirmation

The final step in submitting form contents is the processing by the appropriate software. Because forms have many uses, they require different types of processing—for example, searching for information in a database, adding information to a database, or transforming in some way the data that was submitted. In any case, form submission and processing always requires an acknowledgement to the user, which confirms that the information was submitted, and which may also contain the results of the submission. For the Nomad Ltd checkout system, Cleveland's department is using **e-commerce software**, which can bill the user's credit card for the amount of the order, and send instructions to the company warehouse to ship the purchased items to the appropriate address. When the order is entered into the Nomad Ltd system, the software will instruct the Web server to display the confirmation Web page that Grace created, which is shown in Figure D-15.



FIGURE D-14: Web page front end and back end

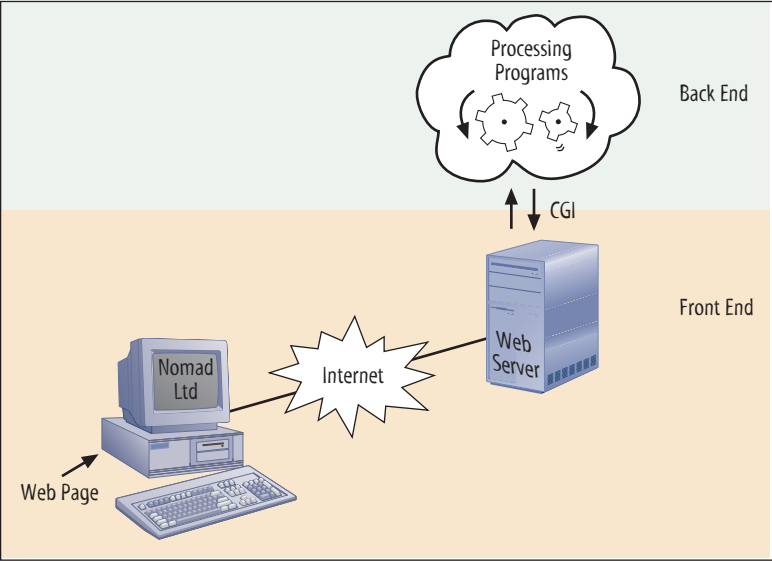


FIGURE D-15: Nomad Ltd order confirmation Web page

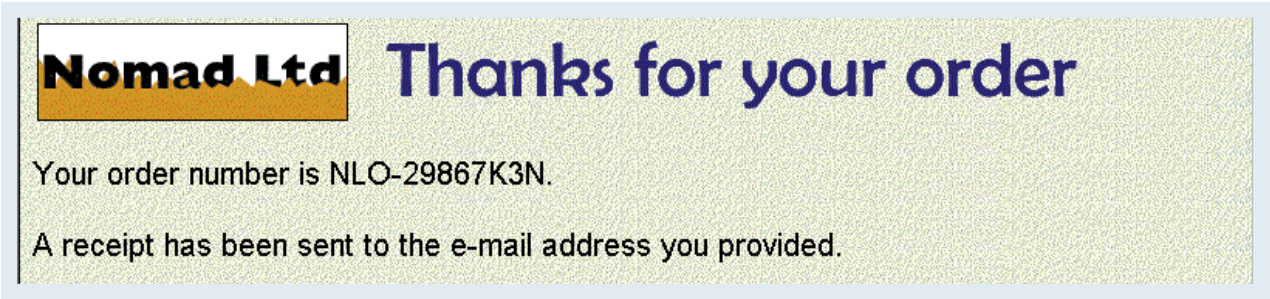


TABLE D-5: <FORM> tag attributes for submitting information

attribute	description	example
METHOD	indicates how the data entered by the user will be submitted to the server; value may be “get” or “post”	METHOD=“get”
ENCTYPE	specifies how the browser connects the user responses on the form before submitting them to the server	ENCTYPE=“application/x-www-form-urlencoded”
ACTION	indicates the path and name of the processing program that the server must run when the form is submitted	ACTION= “/cgi-bin/ExecMacro/input”

## ► Concepts Review

Label each element of the browser screen shown in Figure D-16

FIGURE D-16

1 ☒ Standard Shipping  
☐ 2 Day Air Shipping  
☐ Overnight Air Shipping

2 **3. Packaging Options**

3 ☐ Gift wrap (\$2.00 per item)  
☐ Remove price tags (free)  
☐ Ship multiple items together

4 **4. Payment Information**

Credit Card:

Card Number:

5 Expiration Date (mm/yyyy):

☐ Billing address same as ship-to address

6

**Match each statement with the HTML INPUT attribute it describes**

- |   |             |
|---|-------------|
| 7. Descriptive text that accompanies a field  | a. Form     |
| 8. The programs that reside on an organization's computer system, and which are responsible for processing the submitted data | b. Field    |
| 9. A form element that allows user input  | c. Label    |
| 10. A group of HTML elements that facilitates interactivity   | d. CGI      |
| 11. The protocol for exchanging information between a Web server and processing software                                      | e. Back end |

**Select the best answer from the list of choices.**

12. Which tag(s) do you use to create a form?
- a. <FORM>..<</FORM>
  - b. <INPUT>
  - c. <SELECT>..<</SELECT>
  - d. <OPTION>
13. Which attribute can you use to preselect an item in a radio button list?
- a. TYPE
  - b. CHECKED
  - c. LENGTH
  - d. ORDER
14. Which of the following form fields requires the VALUE attribute?
- a. Text box
  - b. Radio button
  - c. Checkbox
  - d. Pull-down menu
15. Which tag(s) mark a menu item in a pull-down menu?
- a. <FORM>..<</FORM>
  - b. <INPUT>
  - c. <SELECT>..<</SELECT>
  - d. <OPTION>



16. Which attribute allows you to customize the text on a push button?
- TEXT
  - TYPE
  - NAME
  - VALUE
17. The Web page that users see in their browsers is called the Web site's
- Form.
  - Front end.
  - Back end.
  - Script.
18. E-commerce software is part of which step in form submission?
- Browser packaging
  - CGI
  - Scripting
  - Order processing and confirmation

## ► Skills Review

### 1. Create a text entry field.

- Start your text editor, then open the file HTM D-2.htm, and save it as a text document with the filename search-d.htm.
- Replace the text "`<!-- BEGIN FORM -->`" with "`<FORM NAME="search">`".
- Replace the text "`<!-- END FORM -->`" with "`</FORM>`".
- Replace the text "`<!-- SEARCH TERMS - TEXT ENTRY FIELD -->`" with "`<INPUT TYPE="text" SIZE="50" NAME="searchterms">`".
- Save your work, start your Web browser, then open the file search-d.htm.
- Click in the text box, then type a few words.

### 2. Add radio buttons.

- In your text editor, replace the text "`<!-- RESULTS PER PAGE - RADIO BUTTONS -->`" with "`<INPUT TYPE="radio" NAME="results" VALUE="10" CHECKED>10`", then press [Enter].
- Insert four spaces, type "`<BR>`", then press [Enter].
- Insert four spaces, then type "`<INPUT TYPE="radio" NAME="results" VALUE="20">20`".
- Save your work, click the browser button on the taskbar, reload the form in the browser window, then scroll down if necessary to view the radio buttons.
- Click the "20" radio button, then click the "10" radio button.

**3. Add checkboxes.**

- a. In your text editor, replace the text "`<!-- INCLUDE - CHECKBOXES -->`" with "`<INPUT TYPE='checkbox' NAME='gear' CHECKED>outdoor store`", then press [Enter].
- b. Insert four spaces, type "`<BR>`", then press [Enter].
- c. Insert four spaces, type "`<INPUT TYPE='checkbox' NAME='corp'>corporate information`", then press [Enter].
- d. Insert four spaces, type "`<BR>`", then press [Enter].
- e. Insert four spaces, then type "`<INPUT TYPE='checkbox' NAME='basement'>bargain basement`".
- f. Save your work, click the browser button on the taskbar, reload the form in the browser window, then scroll down if necessary to view the checkbox list.
- g. Click each unchecked checkbox to check it, then click each checkbox again to remove the check marks.

**4. Create a pull-down menu.**

- a. In your text editor, replace the text "`<!-- LANGUAGE - PULL-DOWN MENU -->`" with "`<SELECT NAME='language'>`", then press [Enter].
- b. Insert six spaces, type "`<OPTION VALUE='e'>English`", then press [Enter].
- c. Repeat Step b to enter each of the remaining four options for the language pull-down menu:  
`<OPTION VALUE='s'>Spanish`  
`<OPTION VALUE='f'>French`  
`<OPTION VALUE='j'>Japanese`  
`<OPTION VALUE='g'>German`
- d. Insert four spaces, type "`</SELECT>`", then press [Enter].
- e. Save your work, click the browser button on the taskbar, reload the form in the browser window, then scroll down to view the pull-down menu.
- f. Click the Language pull-down arrow, then click one of the options on the menu that opens.

**5. Add a push button.**

- a. In your text editor, replace the text "`<!-- SUBMIT BUTTON -->`" with "`<INPUT TYPE='submit' VALUE='Search' NAME='search'>`".
- b. Save your work, then click the browser button in the taskbar.
- c. Reload the form in the browser window, then scroll down to view the submit button.
- d. Click the Search button.

## ► Independent Challenges

**1.** You are maintaining a Web site for your computer consulting business, Star Dot Star. In addition to giving basic information to prospective clients, you want to add services to your Web site for your current clients as well. As a first step, you decide to add Web search capabilities to your home page.

To complete this independent challenge:

- a. Start your text editor, open the file HTM D-3.htm, and save it as a text document with the filename sds-d.htm.
- b. Insert the opening and closing `<FORM>` tags, along with any necessary attributes.
- c. Add a text box for search terms, along with a label.
- d. Create a checkbox for selecting the format of search results, and label it “show headers only.”
- e. Create a pull-down menu listing options for number of results per page; include at least five options, and appropriately label the field.
- f. Include a submit button with a label of your choice.
- g. Save your work, open the file in your browser, then test the Web page.



**2.** In your job at the local water department, you have created a Web site that includes information on the local watershed. You want the site to have a feedback form that allows users to submit questions.

To complete this independent challenge:

- a. Start your text editor, open the file HTM D-4.htm, and save it as a text document with the filename mw-d.htm.
- b. Insert the opening and closing `<FORM>` tags, along with any necessary attributes.
- c. Create a pull-down menu with the label “Topic” that lists the following possible topics: Local watershed, Metro Water rates, Water conservation, Other topic.
- d. Add a checkbox that allows users to indicate whether or not they wish to be contacted by Metro Water; be sure to include an appropriate label.
- e. Add an appropriately sized text box with the label “E-mail address.”
- f. Add a large text box with the label “Your question or comment:”.
- g. Create a submit button labeled “Submit.”
- h. Save your work, open the file in your browser, then test the Web page.

**3.** You're expanding an in-store information system for your video store, Film Clips. You created the information system with HTML documents. You want to include a form in the system so customers can tell you about their film preferences and suggest videos they'd like to rent, then you can take their desires into account when you order new stock.

To complete this independent challenge:

- a. Start your text editor, create a new document, and save it as a text document with the filename `fc-d.htm`.
- b. Create a pull-down menu along with an appropriate label that lets a customer select the genre of movie they view the most often (such as drama, action, documentary, or comedy).
- c. Create a series of labeled checkboxes that ask about video technologies the user has at home (for example, VCR, DVD, HDTV, and Other).
- d. Add a labeled text box that allows users to specify additional technologies if they checked "Other."
- e. Add a large labeled text box for users to list movies they'd like to rent from Film Clips.
- f. Save your work, open the file in your browser, then test the Web page.



**4.** One of the popular uses of forms on the Web is search engines. Although many popular search engines exist, each uses a different technology to perform the search. Thus, each search engine's front end is slightly different, depending on the type of information the search engine needs to collect from users. Examining the code for different search engine Web pages can suggest different uses of the form fields with which you're familiar.

To complete this independent challenge:

- a. Connect to the Internet and use your browser to go to a search engine, such as one of the following:

*www.google.com*

*www.altavista.com*

*www.yahoo.com*

If you have trouble locating a search engine, go to [www.course.com](http://www.course.com), navigate to the page for this book, click the link for the Student Online Companion, click the link for this unit, and use the links listed there as a starting point for your search.

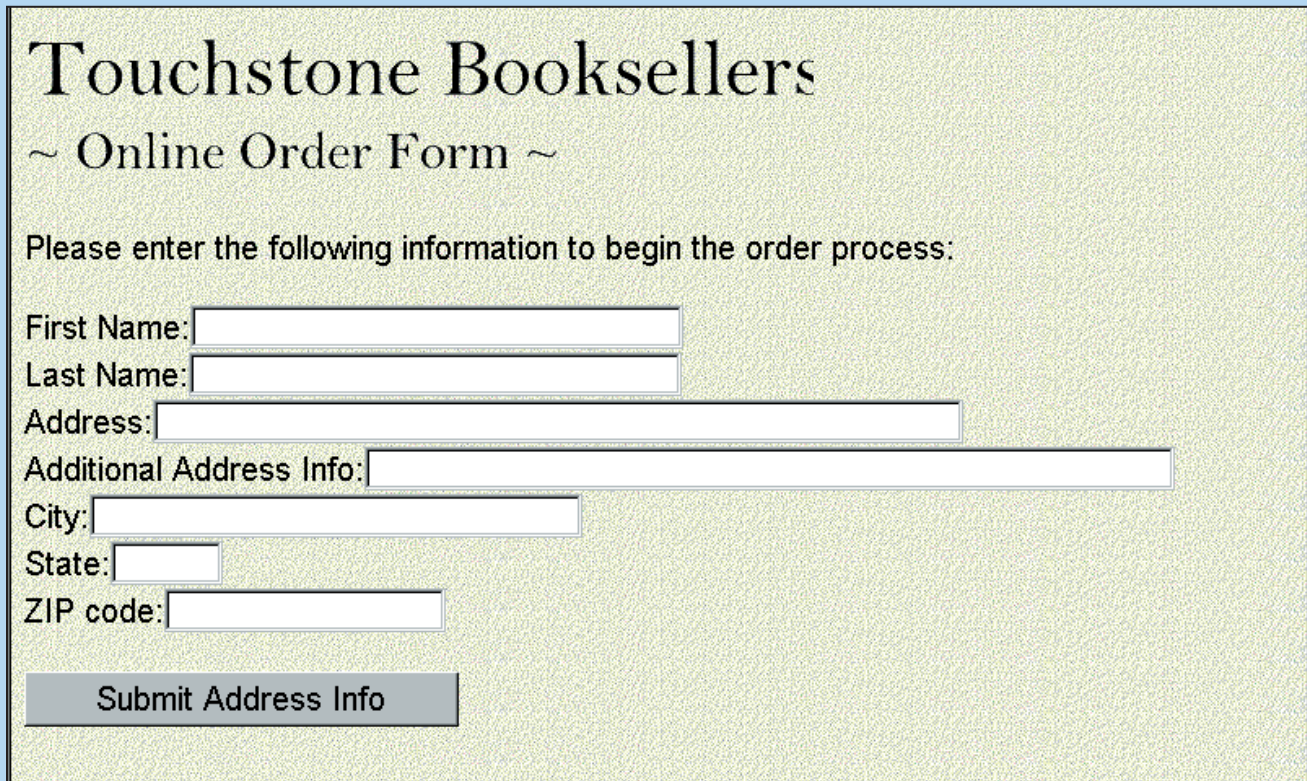
- b. Instead of performing a search, click View, then click Source (or Page Source) in your browser to view the HTML code for the Web page.
- c. Locate the opening `<FORM>` tag, then study the form's field tags. Note any interesting, unusual, or useful implementations of the tags that you've learned; write down tags or attributes within the form code that are unfamiliar to you.
- d. Repeat steps a-c two more times, for a total of three browsers.
- e. Use a search engine to research one of the unfamiliar tags or attributes you noted.
- f. Write a paragraph for each browser, summarizing its new, interesting, or unfamiliar code elements. Write an additional paragraph that gives an overview of the new tag or attribute you researched.



## ► Visual Workshop

As part of your job at Touchstone Books, you've been working to create a Web presence for the store. The store's owner has been researching e-commerce software, and wants you to create a front end form that would allow Web users to order books online. Open the file HTM D-5.htm in your text editor, save it as a text document called tsb-d.htm, then add a form to the Web page so it matches the page shown in Figure D-17.

FIGURE D-17



The image shows a web form titled "Touchstone Booksellers ~ Online Order Form ~". The form is set against a light green textured background. It contains several text input fields for user information: First Name, Last Name, Address, Additional Address Info, City, State, and ZIP code. A gray button labeled "Submit Address Info" is positioned at the bottom of the form.

Touchstone Booksellers  
~ Online Order Form ~

Please enter the following information to begin the order process:

First Name:

Last Name:

Address:

Additional Address Info:

City:

State:

ZIP code: